

# A Pliable Index Coding Approach to Data Shuffling

Linqi Song and Christina Fragouli

## Abstract

A promising research area that has recently emerged, is on how to use index coding to improve the communication efficiency in distributed computing systems, especially for data shuffling in iterative computations. In this paper, we posit that pliable index coding can offer a more efficient framework for data shuffling, as it can better leverage the many possible shuffling choices to reduce the number of transmissions. We theoretically analyze pliable index coding under data shuffling constraints, and design an hierarchical data-shuffling scheme that uses pliable coding as a component. We find benefits up to  $O(ns/m)$  over index coding, where  $ns/m$  is the average number of workers caching a message, and  $m$ ,  $n$ , and  $s$  are the numbers of messages, workers, and cache size, respectively.

## I. INTRODUCTION

A promising research area that has recently emerged, is on how to use coding techniques to improve the communication efficiency in distributed computing systems [1], [2], [3]. In particular, index coding has been proposed to increase the efficiency of data shuffling, that can form a major communication bottleneck for big data applications [1], [3], [4]. In this paper, we posit that using pliable index coding, as opposed to the traditional index coding, would offer a more efficient framework and higher benefits for data shuffling.

In data shuffling a master node has  $m$  messages and is connected through a broadcast channel to  $n$  worker nodes. Each worker is equipped with a cache that can store  $s_i$  messages. Shuffling occurs in iterations, where in each iteration we need to refresh the data the workers have, with a random selection of  $s_i$  out of  $m$  messages from the master node. Application examples include distributed machine learning, where data shuffling updates the training data in workers [1], and

L. Song and C. Fragouli are with the Department of Electrical Engineering, University of California, Los Angeles. Email: {songlinqi, christina.fragouli}@ucla.edu.

mobile cloud gaming systems where each iteration equips the users with new attributes, e.g., new maps [5].

The index coding formulation is as follows [6]. The messages that worker nodes have from previous iterations form side information. The master node randomly interleaves the  $m$  messages and allocates some specific  $s_i$  messages to each worker. This is equivalent to an index coding problem, where each worker makes some specific  $s_i$  requests. Index coding aims to find the smallest amount of broadcast transmissions to satisfy all requests. However, the index coding problem has been shown to be NP-hard and in the worst case may require  $\Omega(n)$  transmissions [6]. For random graph instances it almost surely requires  $\Theta(n/\log(n))$  transmissions [7], [8].

We posit that the framework of pliable index coding could be a better fit for such applications. Our observation is that, when shuffling, we do not need to pre-specify the new messages a worker gets; we only need the worker to get messages that are new and are randomly selected from the original messages. For example, to train a classification model in a distributed system, large volume of data instances can be randomly distributed into  $n$  worker nodes in tens of millions of ways, not necessarily in a specific way. Pliable index coding assumes that the workers are pliable and are happy to receive any new messages they do not already have [9]. This degree of freedom enable us to design more efficient coding and transmission schemes to realize semi-random shuffling. Indeed, pliable index coding requires in the worst case  $O(\log^2(n))$ , an exponentially smaller number of transmissions than index coding, and these benefits can be achieved in polynomial time [9], [10].

In this paper, we make two main contributions.

First, we analyze how pliable index coding would perform under the constraints of data shuffling. In particular, when data shuffling, we want each message to go to at most a specific number of workers, say  $c$ , to achieve an unbiased data distribution that looks “random-like”. We capture this by imposing the constraint that each message can be used to satisfy at most  $c$  clients. That is, each client is happy to receive any message she does not have, but at most  $c$  clients can receive the same message. We show that even if  $c = 1$ , i.e., each message can satisfy at most one client, we can still achieve  $O(n)$  benefits over index coding in some cases; this is because, we still have the freedom to select any of the  $O(n!)$  interleaved versions of requests that lead to the smallest number of transmissions. We prove that the constrained pliable index coding problem is NP-hard. We show that for random instances, the optimal code length is almost surely upper bounded by  $O(\min\{\frac{n}{c \log(n)}, \frac{n}{\log(m)}\})$  for  $c = o(\frac{n^{1/7}}{\log^2(n)})$  and  $O(\min\{\frac{n}{c} + \log(c), \frac{n}{\log(m)}\})$  for

$$c = \Omega\left(\frac{n^{1/7}}{\log^2(n)}\right).$$

Second, we design a hierarchical transmission scheme for data shuffling that utilizes constrained pliable index coding as a component. We introduce a Hamming distance measure to quantify the shuffling performance, and show that our scheme can achieve benefits  $O(ns/m)$ , in terms of transmissions over index coding, with linear encoding complexity at the master node, where  $s$  is the cache size and  $ns/m$  is the average number of workers that cache each message.

## II. CONSTRAINED PLIABLE INDEX CODING

### A. Formulation

We consider a server with  $m$  messages  $b_1, b_2, \dots, b_m$  in a finite field  $\mathbb{F}_q$  and  $n$  clients. Throughout the paper, we will use  $[y]$  ( $y$  is a positive integer) to denote the set  $\{1, 2, \dots, y\}$  and use  $|Y|$  to denote the cardinality of set  $Y$ . We will interchangeably use  $b_j$  and message  $j \in [m]$  to refer to messages. Each client has as side information some subset of the messages, indexed by  $S_i \subseteq [m]$ . Client  $i \in [n]$  has requested any one of the remaining messages, indexed by  $R_i = [m] \setminus S_i$ . We term this set  $R_i$  the request set.

*c-constraint:* We require that a message  $j$  is stored by at most  $c$  clients. We call such a problem  $c$ -constrained pliable index coding and denote it by  $(m, n, \{R_i\}_{i \in [n]}, c)$ .

*Bipartite Graph Representation:* In the bipartite graph, on one side the vertices correspond to messages and on the other side to clients; we connect clients to the messages they *do not* have, i.e., client  $i$  connects to the messages in  $R_i$  [11].

*Linear Encoding:* The server makes  $L$  broadcast transmissions  $x_1, x_2, \dots, x_L$  over a noiseless channel. Each  $x_l$  is a linear combination of  $b_1, \dots, b_m$ , namely,  $x_l = a_{l1}b_1 + a_{l2}b_2 + \dots + a_{lm}b_m$ , where  $a_{lj} \in \mathbb{F}_q$  is the encoding coefficient. We refer to the number of transmissions,  $L$ , as the *code length* and to the  $L \times m$  coefficient matrix  $\mathbf{A}$  with entries  $a_{lj}$  as the *coding matrix*. In matrix form, we can write

$$\mathbf{x} = \mathbf{A}\mathbf{b}, \tag{1}$$

where  $\mathbf{b}$  and  $\mathbf{x}$  are vectors that collect the original messages and encoded transmissions, respectively.

*Linear Decoding:* Given  $\mathbf{A}$ ,  $\mathbf{x}$ , and  $\{b_j | j \in S_i\}$ , each client  $i$  needs to solve the linear equation (1) to get a unique solution of  $b_{j_i}$ , for some  $j_i \in R_i$ . We say that client  $i$  is satisfied if she stores the decoded message  $b_{j_i}$  and  $b_{j_i}$  is decoded and stored by at most  $c$  clients. Clearly,

client  $i$  can remove from the transmissions her side information messages, i.e., to recover  $x_l^{(i)} = x_l - \sum_{j \in S_i} a_{lj} b_j$  from the  $l$ -th transmission. As a result, client  $i$  only needs to solve

$$\mathbf{A}_{R_i} \mathbf{b}_{R_i} = \mathbf{x}^{(i)}, \quad (2)$$

to retrieve a message  $b_{j_i}$  she does not have, where  $\mathbf{A}_{R_i}$  is the sub-matrix of  $\mathbf{A}$  with columns indexed by  $R_i$ ;  $\mathbf{b}_{R_i}$  is the message vector with elements indexed by  $R_i$ ; and  $\mathbf{x}^{(i)}$  is a  $L$ -dimensional column vector with elements  $x_l^{(i)}$ .

Building on results in [12], we have the following decoding criterion. We use  $\mathbf{a}_j$  to denote the  $j$ -th column of matrix  $\mathbf{A}$  and use  $\text{span}\{\mathbf{a}_{j'} | j' \in R_i \setminus \{j\}\} = \{\sum_{j' \in R_i \setminus \{j\}} \lambda_{j'} \mathbf{a}_{j'} | \lambda_{j'} \in \mathbb{F}_q\}$  to denote the linear space spanned by columns of  $\mathbf{A}$  indexed by  $R_i$  other than  $j$ .

**Lemma 1.** *In a constrained pliable index coding problem  $(m, n, \{R_i\}_{i \in [n]}, c)$ , a coding matrix  $\mathbf{A}$  can satisfy all clients if and only if there exist messages  $j_1, j_2, \dots, j_n \in [m]$ , one for each client, where no single message is repeated more than  $c$  times, i.e.,  $j_{i_1} = j_{i_2} = \dots = j_{i_{c+1}}$  does not hold for any combination of  $c+1$  clients  $i_1, i_2, \dots, i_{c+1} \in [n]$ , such that the matrix  $\mathbf{A}$  satisfies*

$$\mathbf{a}_{j_i} \notin \text{span}\{\mathbf{a}_{j'} | j' \in R_i \setminus \{j\}\}, \forall i \in [n]. \quad (3)$$

Our goal is to construct the coding matrix  $\mathbf{A}$  with the minimum code length  $L$ . Note that the  $c$ -constraint significantly changes the pliable index coding problem. For example, assume we have  $m$  messages and  $n$  clients with no side information; then pliable index coding requires 1 transmission, while constrained pliable index coding needs  $n/c$  transmissions to satisfy all clients.

## B. Main Results

*1) Benefits Over Index Coding:* Clearly, the larger the value of  $c$ , the more benefits we expect constrained pliable index coding to have over index coding (for  $c = n$  we have exponential benefits [9], [10]). We here provide an example to show that it is possible to have benefits of  $O(n)$  even when  $c = 1$ , i.e., each message can satisfy at most one client, as is the case in index coding. This equivalently shows that, if we are allowed to “interleave the demands” in index coding, we can gain  $O(n)$  in terms of the number of transmissions.

We construct the following 1-constrained pliable coding instance with  $n$  messages and  $n$  clients. Client  $i \in [n/2]$  requests any of the messages 1 to  $n/2$  and  $n/2 + i$ , i.e.,  $R_i =$

$\{1, 2, \dots, n/2, n/2 + i\}$ , for  $i \in [n/2]$ . Client  $i \in [n] \setminus [n/2]$  requests any of the messages  $n/2 + 1$  to  $n$  and  $i - n/2$ , i.e.,  $R_i = \{i - n/2, n/2 + 1, n/2 + 2, \dots, n\}$ , for  $i \in [n] \setminus [n/2]$ . All messages not in the request set form side information.

For index coding, if client  $i$  requests message  $i$  and has the same side information as above, then we need at least  $n/2$  transmissions, since the first  $n/2$  clients do not have the first  $n/2$  messages as side information. In contrast, 1-constrained pliable index coding only requires 2 transmissions. Indeed, we can enable client  $i \in [n/2]$  to decode the message  $n/2 + i$ , by making the transmission  $b_{n/2+1} + b_{n/2+2} + \dots + b_n$ , since each client  $i \in [n/2]$  has all messages indexed by  $[n] \setminus ([n/2] \cup \{n/2 + i\})$  as her side information. Similarly, we can enable client  $i \in [n] \setminus [n/2]$  to decode the message  $i - n/2$  by making the transmission  $b_1 + b_2 + \dots + b_{n/2}$ .

2) *Constrained Pliable Index Coding is NP-hard*: It suffices to show that 1-constrained pliable index coding is NP-hard.

**Theorem 1.** *For a 1-constrained pliable index coding problem, deciding if the optimal code length*

- $L = 1$  is in  $P$ .
- $L = 2$  is NP-complete.

The  $L = 1$  case is easy to see: if one transmission can make each client to receive a distinct message, then the server needs to linearly combine exactly  $n$  messages, one for each client. Client  $i$  can decode a message  $b_j$ ,  $j \in R_i$ , only if all other  $n - 1$  messages are in her side information set. A greedy approach enables to test whether such  $n$  messages exist can be tested in polynomial time. For  $L = 2$ , we use a reduction from the graph coloring problem, see Appendix A for the complete theorem proof.

3) *Performance Over Random Instances*: We consider a random bipartite graph instance, denoted by  $B(m, n, c, p)$ , or  $B$  for short, where there are  $m$  messages,  $n$  clients, each message can be recovered and stored by at most  $c$  clients, and each client is connected with a message with probability  $p$  (clients have as side information all the messages they are not connected to). We assume that  $p$  is a fixed constant, while  $c = c(n)$  and  $m = m(n) \geq n$  could be changing with  $n$ .

Theorem 2 summarizes our main result; we provide a proof outline in the rest of the section, and a complete proof in Appendix B.

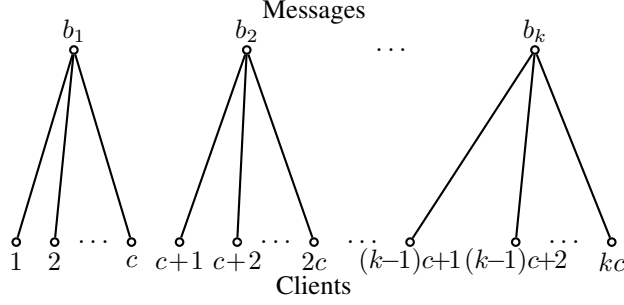


Fig. 1: In a  $k$ -pattern, one transmission satisfies  $kc$  clients.

**Theorem 2.** *The number of broadcast transmissions for random graph instance  $B(m, n, c, p)$  is almost surely upper bounded by*

- $O(\min\{\frac{n}{c \log(n)}, \frac{n}{\log(m)}\})$ , for  $c = o(\frac{n^{1/7}}{\log^2(n)})$ ; and
- $O(\min\{\frac{n}{c} + \log(c), \frac{n}{\log(m)}\})$ , for  $c = \Omega(\frac{n^{1/7}}{\log^2(n)})$ .

Our proof is constructive: we design a transmission scheme, and show that it achieves this performance. To do so, we first define a  $k$ -pattern to be an induced subgraph that consists of  $k$  message vertices,  $kc$  client vertices, each of the  $k$  messages is connected with  $c$  clients, and each of the  $kc$  client is connected with only one of the messages (see Fig. 1). A  $k$ -pattern enables with a single broadcast transmission to satisfy all the  $kc$  clients. We then find values  $\mathbb{K} = \mathbb{K}(m, n)$ ,  $m'$  and  $n'$  for which almost surely a  $\mathbb{K}$ -pattern exists in every induced subgraph  $B'$  of  $B$  with  $m'$  message vertices and  $n'$  client vertices, i.e.,

$$\Pr\{\exists B', s.t., B' \text{ contains no } \mathbb{K}\text{-patterns}\} = o(1). \quad (4)$$

The transmission scheme proceeds as follows. If there are more than  $n'$  clients in the original graph  $B$ , we pick a  $\mathbb{K}$ -pattern and make one transmission. We remove the satisfied clients and the used messages. If there are less than  $n'$  clients, we use at most  $n'$  transmissions to satisfy the remaining clients. Hence, we almost surely need  $\frac{n}{\mathbb{K}c} + n'$  transmissions.

To minimize  $\frac{n}{\mathbb{K}c} + n'$ , we want  $n'$  to be small and  $\mathbb{K}$  to be large. However, by decreasing  $n'$  we also decrease the values of  $\mathbb{K}$  that satisfy (4). Hence, we need to balance the sizes of  $n'$  and  $\mathbb{K}$ ; we use different values depending on how  $m$ ,  $n$ , and  $c$  are related.

In the following, we consider  $c = o(\frac{n^{1/7}}{\log^2(n)})$ ,  $m' = \frac{m}{\log(n)}$ ,  $n' = \frac{n}{c \log(n)}$ ,  $\mathbb{K} = \lfloor \frac{1}{\log(1/\bar{p})} [\log(n) - 3 \log \log(n) - 3 \log(c) + 2 \log \log(\frac{1}{\bar{p}})] \rfloor = \Theta(\log(n))$ , and  $\bar{p} = \min\{p, 1 - p\}$ , and show that we

can satisfy the condition in (4):

$$\begin{aligned}
& \Pr\{\exists B', \text{s.t., } B' \text{ contains no } \mathbb{K}\text{-patterns}\} \\
& \leq \binom{m}{m'} \binom{n}{n'} \exp\left(-\frac{\bar{p}^2 \log^{12}(1/\bar{p})mn}{8c^7 \log^{14}(n)}\right) \\
& \leq 2^{m+n} \exp\left(-\frac{\bar{p}^2 \log^{12}(1/\bar{p})mn}{8c^7 \log^{14}(n)}\right) = o(1).
\end{aligned} \tag{5}$$

The first inequality follows from the following lemma.

**Lemma 2.** *The probability that the subgraph  $B'$  contains no  $\mathbb{K}$ -pattern is no more than  $\exp(-\frac{\bar{p}^2 \log^{12}(1/\bar{p})mn}{8c^7 \log^{14}(n)})$ .*

For a complete proof of Lemma 2, see Appendix B. The outline of the proof is as follows. Let  $Y_k^{B'}$  to be the number of  $k$ -patterns over a subgraph  $B'$ . The expected number of  $k$ -patterns equals:

$$\mathbb{E}[Y_k^{B'}] = \binom{m'}{k} \binom{n'}{kc} \binom{kc}{c, c, \dots, c} p^{kc} (1-p)^{kc(k-1)}.$$

It is easy to see that  $\mathbb{E}[Y_k^{B'}]$  is decreasing with  $k$ , assuming other parameters fixed. We define  $k_0^{B'}$  to be the maximum integer such that  $\mathbb{E}[Y_{k_0^{B'}}^{B'}] \geq 1$ , i.e.,  $k_0^{B'} = \max\{k | \mathbb{E}[Y_k^{B'}] \geq 1\}$ . It is not hard to calculate that

$$\begin{aligned}
k_0^{B'} &= \frac{1}{\log(1/(1-p))} [\log(n) + \frac{\log(m)}{c} - 2 \log(c) - \log \log(n) \\
&\quad - \frac{\log \log(n)}{c} - \frac{\log[\log(n) + \log(m)/c - 2 \log(c)]}{c} \\
&\quad - \frac{\log \log(1/(1-p))}{c} + \log(p)] + O(1)
\end{aligned}$$

and

$$\mathbb{E}[Y_{k_0^{B'}-3}^{B'}] \geq \left(\frac{n}{ec^2 \log(n)}\right)^{3c(1+o(1))} \left(\frac{m}{\log(n)}\right)^{3(1+o(1))} \tag{6}$$

Hence, we see that  $\mathbb{K} \leq k_0^{B'} - 3$ . Then we use an “edge-exposure” process [13] to form a martingale and use the following Azuma’s inequality to bound the probability in Lemma 2:

$$\Pr\{B' \text{ contains no } \mathbb{K}\text{-pattern}\} \leq \exp\left(-\frac{\mathbb{E}^2[X]}{2m'n'}\right), \tag{7}$$

where  $X$  is the maximum number of  $\mathbb{K}$ -patterns in  $B'$  such that no pair of them share a same message-client pair (i.e., any two  $\mathbb{K}$ -patterns either have no common message vertices or client vertices or both), and it is shown that

$$\mathbb{E}[X] \geq \frac{\bar{p} \log^6(\frac{1}{\bar{p}})mn}{2c^4 \log^8(n)}. \tag{8}$$

### III. APPLICATION TO DISTRIBUTED COMPUTING

#### A. Model and Performance Metric

Consider a distributed computing system, with one master node with  $m$  messages  $b_1, b_2, \dots, b_m$  and  $n$  worker nodes. Each worker  $i \in [n]$  is equipped with a cache of size  $s$ . The system solves a computational problem  $x = f(b_1, b_2, \dots, b_m)$  in iterations, where at iteration  $t$ : the master broadcasts the current estimate  $x^{t-1}$  to all workers; workers perform local computations and send to the master their new estimate  $x_i^t$ ; the master node combines local estimates to get an updated estimate; he then performs data shuffling. In data shuffling, the master node makes broadcast transmissions (that may be encoded) to the workers and each worker replaces some of the old messages with the new messages that she can decode.

We use a Hamming distance metric to evaluate the performance of data shuffling algorithms. Let the indicator vector for the messages stored at worker node  $i$  in time period  $t$  be  $z_i^t \in \{0, 1\}^m$ , where the  $j$ -th bit of  $z_i^t$  takes value 1 if message  $b_j$  is in the cache of worker node  $i$  in time period  $t$  and 0 otherwise. The Hamming distance between two indicators  $z$  and  $z'$ , denoted by  $H(z, z')$ , is the number of positions where the entries are different for  $z$  and  $z'$ . We define the Hamming distance of a shuffling scheme as the average Hamming distance across time and worker nodes  $H \triangleq \frac{1}{\binom{Tn}{2}} \sum_{1 \leq t, t' \leq T, (i, i') \in [n]^2, (t, i) \neq (t', i')} \mathbb{E}[H(z_i^t, z_{i'}^{t'})]$ , where  $T$  denotes the number of iterations.

#### B. Data Shuffling Scheme

1) *Hierarchical Structure*: We partition the messages into  $m/m_1$  groups so that each group  $g$  contains  $m_1$  disjoint messages. In our scheme, each worker  $i$  gets allocated messages from groups indexed by a set  $D(i)$ ; each group  $g$  allocates messages to workers indexed by a set  $N(g)$ . We can represent this relationship using a bipartite graph: at one side there are  $m/m_1$  groups, and at the other side there are  $n$  workers; there is a connection between worker  $i$  and group  $g$  if and only if worker  $i$  caches messages from group  $g$ , i.e.,  $g \in D(i)$ ; the degree of the worker node  $i$  is  $|D(i)|$  and of the group node  $g$  is  $|N(g)|$ . This structure is maintained for all iterations.

In order to have a large Hamming distance  $H$ , we can impose the constraint that  $|D(i) \cap D(i')| \leq 1$  for any two worker nodes  $i$  and  $i'$ , namely, they have common messages in no more than one group. Moreover, to balance the messages cached in different worker nodes,



we would like that  $|N(g)|$  is the same for all groups. We thus select for our scheme to use  $|D(i)| = \frac{s}{m_1(1-1/r)}$ , and  $|N(g)| = \frac{ns}{m(1-1/r)}$ , where the design parameter  $1 \leq r \leq m_1$  takes integer values.

Note that because of the requirement that the same message can be decoded and stored by at most  $c$  workers, we need that no more than  $rc$  workers cache messages in group  $g$ , i.e.,  $|N(g)| \leq cr$ . For example, for  $c = 1$ , then at most  $r$  workers can be in  $N(g)$ , each one of them with  $m_1(1 - 1/r)$  cached messages from this group.

2) *Transmissions*: We proceed as follows.

- Initialization: the cache of worker  $i$  is filled with uniformly at random selected  $m_1(1 - 1/r)$  messages from each group in  $D(i)$ , thus in total  $s$  messages.
- Iteration  $t$ : the master makes  $m/m_1$  broadcast transmissions, one for every group. For each group  $g$ , the master selects uniformly at random  $r$  messages in the group and transmits their linear combination, say  $b_{j_1} + b_{j_2} + \dots + b_{j_r}$ . From the following Lemma 3, every worker in  $N(g)$  can decode a new message with probability at least  $1/e$ . The workers who can decode a new message store it in their cache and discard an old message; they select the old message to discard uniformly at random from the messages in their cache that are also contained in the broadcast transmission, i.e., one from  $\{b_{j_1}, b_{j_2}, \dots, b_{j_r}\}$ .

**Lemma 3.** *A worker with  $m_1(1 - 1/r)$  cached messages from group  $g$  that receives a linear combination  $b_{j_1} + b_{j_2} + \dots + b_{j_r}$  of  $r$  messages uniformly at random selected from  $g$ , can decode a message she does not have with probability at least  $1/e$ .*

*Proof.* Without loss of generality, assume the worker has cached the messages  $1, 2, \dots, m_1(1 - 1/r)$  and requires a new message from the remaining  $m_1/r$  messages. The probability that there is exactly one message in the  $r$  data pieces  $b_{j_1}, b_{j_2}, \dots, b_{j_r}$  selected from the last  $m_1/r$  data pieces is lower bounded by

$$\begin{aligned}
 p_1 &\triangleq \Pr\{\text{The worker can decode a new message}\} \\
 &\geq \frac{\binom{m_1/r}{r} \binom{m_1(1-1/r)}{r-1}}{\binom{m_1}{r}} \\
 &= \frac{(m_1 - \frac{m_1}{r})(m_1 - \frac{m_1}{r} - 1) \dots (m_1 - \frac{m_1}{r} - r + 2)}{(m_1 - 1)(m_1 - 2) \dots (m_1 - r + 1)} \\
 &\geq \left(\frac{m_1 - \frac{m_1}{r} - r + 2}{m_1 - r + 1}\right)^{r-1} = \left(1 - \frac{m_1 - r}{r(m_1 - r + 1)}\right)^{r-1} \\
 &\geq \left(1 - \frac{1}{r}\right)^{r-1} \geq \frac{1}{e}
 \end{aligned}$$

□

3) *Algorithm Performance:* We here theoretically evaluate properties of the proposed algorithm.

- **Communication cost.** Each data shuffling phase requires  $m/m_1$  broadcast transmissions.
- **Satisfying  $c$ -constraint.** As at most  $rc$  workers have cached messages from group  $g$ , from Lemma 3, we can see that on average at most  $rcp_1$  (for some fixed  $1 > p_1 \geq 1/e$ ) workers can update their cache with a new message during one transmission. Because we uniformly at random select which  $r$  messages to encode, each message can be decoded by  $cp_1$  workers on average. Hence, on average, the  $c$ -constraint is satisfied. Note that this scheme allows us to maintain the randomness property for workers in  $N(g)$  (see Appendix C).

- **Hamming distance.** Between the caches of any two workers the Hamming distance is at least  $2(s - m_1 + m_1/r)$ , since any two workers have common messages from at most one group.

Next, we evaluate the Hamming distance across iterations for the same worker. We first consider the Hamming distance  $H|_g$  only corresponding to the messages of a specific group  $g$ . The average Hamming distance across all iterations is at least the average Hamming Distance between two consecutive iterations (see Appendix C). Hence, the average Hamming distance  $H|_g$  can be lower bounded by:

$$H|_g \geq 0 \cdot (1 - \frac{1}{e}) + 2 \cdot \frac{1}{e} = 2/e. \quad (9)$$

We then consider the average Hamming distance across all the groups in  $D(i)$ . Since  $|D(i)| = \frac{s}{m_1(1-1/r)}$ , this is at least  $\frac{s}{m_1(1-1/r)} 2/e = \frac{2s}{em_1(1-1/r)}$ . Therefore, on average  $H \geq \min\{\frac{2s}{em_1(1-1/r)}, 2(s - m_1 + m_1/r)\}$ .

- **Comparison to Index Coding.** Index coding may require in the worst case  $\Omega(n)$  broadcast transmissions and  $\Theta(n/\log(n))$  for random graph instances to update one message in each cache, and thus  $\Omega(ns/em_1(1 - 1/r))$  (in the worst case) and  $\Theta(ns/em_1(1 - 1/r) \log(n))$  (for random graph instances) broadcast transmissions in each data shuffling iteration to guarantee a Hamming distance of  $\frac{2s}{em_1(1-1/r)}$  across time. Using our proposed scheme, we need  $m/m_1$  transmissions to achieve an average Hamming distance of  $2s/em_1(1 - 1/r)$  across time. Let us define  $c_1 = s/m$ , then on average each message is stored on  $c_1 n$  workers. The benefits of our proposed scheme over index coding (i.e., the ratio of the numbers of transmissions for index coding scheme and for our proposed scheme) is  $O(c_1 n)$  (in the worst case) and  $O(\frac{c_1 n}{\log(n)})$  (for

random graph instances). Additionally, finding the optimal index coding solution is NP-hard, while our scheme has linear complexity of encoding.

### C. Experimental Results

We conduct experiments on distributed machine learning over a real data set<sup>1</sup> that aims to detect diseased trees in an image. We train the distributed classification model using a stochastic gradient descent method based on 1000 data instances (messages). We set the number of workers to  $n = 300$  and the cache size to  $s = 10$ . We divide the messages into 50 groups, with 20 messages in each. We set the parameter  $r = 1$ , i.e., each worker has cached messages in 1 group. We carry out experiments by comparing our hierarchical pliable index coding based shuffling against: (i) no shuffling and (ii) shuffling with randomly selected messages. For case (ii), once we randomly select what message to send to each worker, we use two approaches for broadcasting: uncoded broadcast transmissions, and index coding [1], [14]. We implemented index coding using the graph coloring based heuristic approach in [14].

In Fig. 2, we compare the computation performance of our pliable index coding based shuffling scheme with no data shuffling scheme and the data shuffling scheme with randomly selected messages (uncoded shuffling and index coding based shuffling use the same cached messages in each local computation, and only differ in the data shuffling phase). We find that data shuffling improves the performance by 11% as compared to no shuffling; pliable coding shuffling performs very similarly (3% better) to randomized shuffling. In Fig. 3, we compare the number of broadcast transmissions for the data shuffling schemes: uncoded shuffling, index coding based shuffling and pliable index coding based shuffling. We find that our proposed pliable index coding based shuffling scheme saves 49% and 44% in terms of number of transmissions as compared to the uncoded shuffling and to the index coding based shuffling, respectively.

## REFERENCES

- [1] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, “Speeding up distributed machine learning using codes,” *arXiv preprint arXiv:1512.02673*, 2015.
- [2] M. Attia and R. Tandon, “Information theoretic limits of data shuffling for distributed learning,” *arXiv preprint arXiv:1609.05181*, 2016.
- [3] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, “Coded mapreduce,” in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2015, pp. 964–971.

<sup>1</sup><https://archive.ics.uci.edu/ml/datasets/Wilt#>

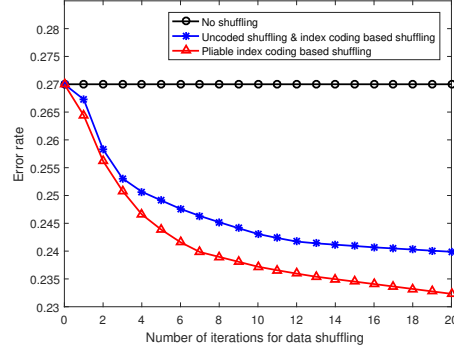


Fig. 2: Comparison of computation performance for data shuffling schemes.

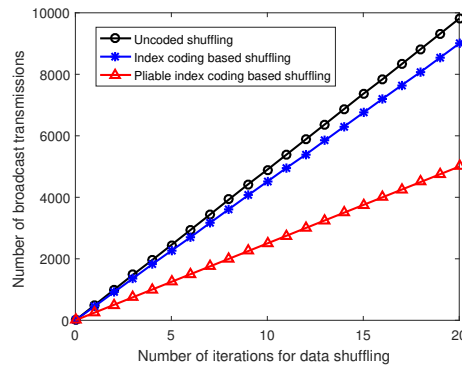


Fig. 3: Comparison of broadcast transmissions for data shuffling schemes.

- [4] M. Chowdhury, M. Zaharia, J. Ma, M. I. Jordan, and I. Stoica, “Managing data transfers in computer clusters with orchestra,” in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4. ACM, 2011, pp. 98–109.
- [5] S. Shi, C.-H. Hsu, K. Nahrstedt, and R. Campbell, “Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming,” in *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011, pp. 103–112.
- [6] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, “Index coding with side information,” *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1479–1494, 2011.
- [7] I. Haviv and M. Langberg, “On linear index coding for random graphs,” in *2012 IEEE International Symposium on Information Theory (ISIT)*, 2012, pp. 2231–2235.
- [8] A. Golovnev, O. Regev, and O. Weinstein, “The minrank of random graphs,” *arXiv preprint arXiv:1607.04842*, 2016.
- [9] S. Brahma and C. Fragouli, “Pliable index coding,” *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6192–6203, 2015.
- [10] L. Song and C. Fragouli, “A polynomial-time algorithm for pliable index coding,” *arXiv preprint arXiv:1610.06845*, 2016.
- [11] S. Brahma and C. Fragouli, “Pliable index coding,” in *2012 IEEE International Symposium on Information Theory (ISIT)*, 2012, pp. 2251–2255.
- [12] L. Song and C. Fragouli, “Content-type coding,” in *2015 International Symposium on Network Coding (NetCod)*, 2015, pp. 31–35.

- [13] B. Bollobás, *Random graphs*. Cambridge Studies in Advanced Mathematics 73, 2001.
- [14] M. A. R. Chaudhry and A. Sprintson, “Efficient algorithms for index coding,” in *IEEE INFOCOM Workshops*, 2008, pp. 1–4.
- [15] R. M. Karp, “Reducibility among combinatorial problems,” in *Complexity of computer computations*. Springer, 1972, pp. 85–103.
- [16] R. Motwani and P. Raghavan, *Randomized algorithms*. Chapman & Hall/CRC, 2010.
- [17] A. Frieze, “Perfect matchings in random bipartite graphs with minimal degree at least 2,” *Random Structures and Algorithms*, vol. 26, no. 3, pp. 319–358, 2005.

## APPENDIX A

### PROOF OF THEOREM 1

In this appendix, we prove Theorem 1.

#### A. Deciding if Optimal $L = 1$ is in P

We first show that deciding if the optimal code length equals 1 is in P. To see this, we notice that if one transmission can make each client to receive a distinct message, then the server needs to encode exact  $n$  messages for the transmission, one for each client. For a client  $i$ , if it can decode a message  $b_j$ ,  $j \in R_i$ , then all other  $n - 1$  messages must be in its side information set following from the decoding criterion. Similarly, any one of the  $n$  messages for encoding is in the side information set of  $n - 1$  clients and requested by the remaining one client. Hence, in the bipartite graph representation, if and only if we can find  $n$  message vertices, such that each one has degree 1 and is connected to a different client vertex, then the optimal code length is 1. This can be tested by going over all message vertices, which runs in polynomial time.

#### B. Deciding if Optimal $L = 2$ is NP-complete

We next show that deciding if optimal code length equals 2 is NP-complete. To prove this, we first introduce another NP-complete problem.

**Definition 1** (Distinct Labeling Problem). *We are given a universal set  $U = \{1, 2, \dots, u\}$  with  $|U| = u$  elements, a fixed set of  $\Pi$  labels  $\{1, 2, \dots, \Pi\}$ , and a collection of size 3 subsets of  $U$ , i.e.,  $\mathcal{S} \subseteq 2^U$  and  $|S| = 3$  for any  $S \in \mathcal{S}$ , where  $2^U$  is the power set of  $U$ . The distinct labeling problem (DL) asks if we can label the elements using  $\Pi$  labels such that every subset in  $\mathcal{S}$  contains elements of 3 different labels. For short, we call it  $\Pi$ -DL problem for such a distinct labeling problem with  $\Pi$  labels.*

**Lemma 4.**  *$\Pi$ -DL problem is NP-complete for  $\Pi \geq 3$ .*

*Proof.* It is easy to see that the  $\Pi$ -DL problem is in NP. We next show that we can use a polynomial time reduction from the graph coloring problem (a.k.a., chromatic number) to the  $\Pi$ -DL problem.

We reiterate the well-known decision version of graph coloring problem as follows [15]: it is NP-complete to decide whether the vertices of a given graph  $G(V, E)$  can be colored using a fixed  $\Pi \geq 3$  colors, such that no two neighboring vertices share the same color.

We perform the following mapping. We map each vertex in  $V$  and each edge in  $E$  as the universal set with  $|U| = |V| + |E|$  elements. We map an edge  $e \in E$  together with the two endpoints  $x_1, x_2$  as a subset, where  $e = \{x_1, x_2\}$ . So there are in total  $|\mathcal{S}| = |E|$  subsets.

We first show that if  $G$  is  $\Pi$ -colorable, then we can find a solution for the  $\Pi$ -DL problem. We can assign a set  $\{1, 2, \dots, \Pi\}$  of colors to the vertices in  $V$ , such that no two neighboring vertices share a same color. When we map to the  $\Pi$ -DL problem, we notice that each edge appears in exact 1 subset, the one corresponding to this edge. Hence, we can use the following labeling scheme: label the elements corresponding to the vertices as the color used in  $\{1, 2, \dots, \Pi\}$ ; and label the edge using any label that is different from its two endpoints. This is a solution for the  $\Pi$ -DL problem.

On the other hand, if we have a solution for the  $\Pi$ -DL problem, we can find a solution for the graph coloring problem. We can label the elements corresponding to vertices using the  $\Pi$  labels. Note that if two vertices  $x_1$  and  $x_2$  are adjacent to each other, i.e.,  $\{x_1, x_2\} \in E$ , then according to the definition of  $\Pi$ -DL problem, these two elements  $x_1$  and  $x_2$  must have different labels. Hence, keep the labels of each vertex element, then we get a  $\Pi$ -coloring of the graph  $G$ .  $\square$

We then prove that deciding if the optimal code length  $L = 2$  is NP-complete for constrained pliable index coding problem over a finite field  $\mathbb{F}_q$ .

First, we observe that we can decide if a given  $2 \times m$  coding matrix  $\mathbf{A}$  can satisfy a constrained pliable index coding instance from our decoding criterion. Indeed, given a coding matrix  $\mathbf{A}$ , one can list the messages a client can decode using the decoding criterion. Then we have a bipartite subgraph representation that has  $n$  clients, some messages, and edges that connect each client with the message she can decode. We only need to check if the maximum matching in such a subgraph equals the number of clients  $n$  using polynomial time. If and only if so, this coding matrix can satisfy the problem instance.

Next, we use a reduction from the  $(q + 1)$ -DL problem defined above to show that the constrained pliable index coding problem is NP-hard. We are given a  $(q + 1)$ -DL problem instance with the universal set  $U = \{1, 2, \dots, u\}$  and a collection of size 3 subsets  $\mathcal{S} \subseteq 2^U$ . We perform the following two mappings.

- For each subset, e.g.,  $S = \{x, y, z\} \in \mathcal{S}$  and  $x, y, z \in U$ , we map into a structure as show in Fig. 4. We map each element in the subset  $S$  as a message vertex and add 3 client vertices

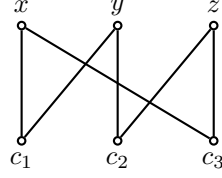


Fig. 4: Mapping a subset into a structure.

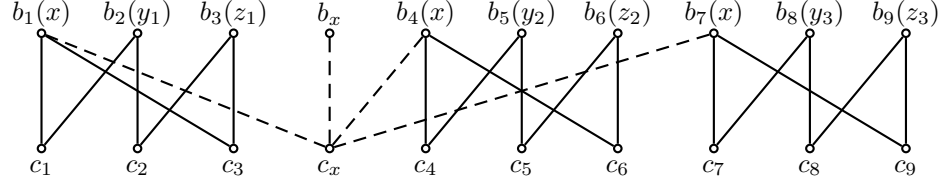


Fig. 5: Connecting the same elements in different subsets.

$c_1, c_2, c_3$  in the constraint pliable index coding problem instance. We connect  $c_1$  to  $x$  and  $y$ , connect  $c_2$  to  $y$  and  $z$ , and connect  $c_3$  to  $z$  and  $x$ .

- For different subsets, if they contain the same element, we connect them using the following structure as shown in Fig. 5. For example, if the subsets  $S_1 = \{x, y_1, z_1\}$ ,  $S_2 = \{x, y_2, z_2\}$ , and  $S_3 = \{x, y_3, z_3\}$  all contain the element  $x$ , we connect a client vertex  $c_x$  to all messages corresponding to  $x$  and another additional message vertex  $b_x$ .

After this mapping, we can see that we construct a constrained pliable index coding instance with  $n = 3|\mathcal{S}| + |U|$  clients and  $m = n$  messages. We want to show that if and only if the  $(q + 1)$ -DL problem outputs a “Yes” answer, a code length 2 coding matrix can satisfy such a problem.

If for a “Yes” instance of  $(q + 1)$ -DL problem, we can find a labeling scheme using  $q + 1$  labels to the elements. In finite field  $\mathbb{F}_q$ , we notice that the maximum number of vectors that are pair-wise independent is  $q + 1$ , e.g.,

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \dots, \begin{bmatrix} 1 \\ q-1 \end{bmatrix}. \quad (10)$$

We consider each label as one of these  $q+1$  vectors. Then for the coding matrix  $\mathbf{A}$ , we can assign the columns the same vector that correspond to the same element in subsets, e.g.,  $b_1, b_4$ , and  $b_7$  in Fig. 5. For the columns corresponding to an element not in subsets, e.g.,  $b_x$ , we assign a different



vector other than the one for the element in subsets. This is a valid coding matrix. Indeed, for 3 messages corresponding to a subset, e.g.,  $b_1$ ,  $b_2$ , and  $b_3$ , they are labeled using different labels from the  $q + 1$ -DL problem solution. Then, the 3 clients corresponding to this subset, i.e.,  $c_1$ ,  $c_2$ , and  $c_3$ , can decode  $b_1$ ,  $b_2$ , and  $b_3$ , respectively, according to the decoding criterion. The client corresponding to an element not in subsets, e.g.,  $c_x$ , can decode the corresponding message not in subsets, i.e.,  $b_x$ , as coding vectors corresponding to messages  $b_1$ ,  $b_4$ , and  $b_7$  are the same and different from the coding vector corresponding to message  $b_x$ .

If for the constrained pliable index coding instance, a length 2 coding matrix  $A$  can make a successful shuffling. Then we notice that client  $c_x$  should be satisfied by message  $b_x$ , since  $b_x$  only connects to  $c_x$  and  $m = n$ , which implies that each message needs to satisfy a client. In this case, the non-zero coding vector corresponding to  $b_x$  is not in the space spanned by other coding vectors corresponding to  $x$  in subsets, i.e.,  $b_1$ ,  $b_4$ , and  $b_7$ . As a result, the space spanned by coding vectors corresponding to the same element in subsets is a one dimensional space, e.g., the space spanned by coding vectors corresponding to  $b_1$ ,  $b_4$ , and  $b_7$ . For clients and messages inside a subset, e.g.,  $b_1$ ,  $b_2$ ,  $b_3$ ,  $c_1$ ,  $c_2$ , and  $c_3$ , there are two ways to satisfy these clients: one is  $b_1$  to  $c_1$ ,  $b_2$  to  $c_2$ ,  $b_3$  to  $c_3$ ; and the other one is  $b_2$  to  $c_1$ ,  $b_3$  to  $c_2$ ,  $b_1$  to  $c_3$ . For both of these two ways, we notice that coding vectors corresponding to  $b_1$ ,  $b_2$ , and  $b_3$  should be pair-wise independent; otherwise, one of the clients cannot decode a new message, e.g., if coding vectors corresponding to  $b_1$  and  $b_2$  are dependent to each other, then  $c_1$  cannot decode any new message. In addition, we observe that there are in total  $q + 1$  1-dimensional subspaces spanned by 2-dimensional vectors over finite field  $\mathbb{F}_q$ , i.e., the spaces spanned by vectors in 10. Therefore, if we assign each space a label, then messages in the same subsets are using different labels and messages in different subsets corresponding to a same element are using the same label, resulting in a solution of the  $q + 1$ -DL problem.

## APPENDIX B

### PROOF OF THEOREM 2

In this appendix, we prove Theorem 2. We consider a random bipartite graph instance, denoted by  $B(m, n, c, p)$ , or  $B$  for short, where there are  $m$  messages,  $n$  clients, each message can be recovered and stored by at most  $c$  clients, and each client is connected with a message with probability  $p$  (clients have as side information all the messages they are not connected to). We

assume that  $p$  is a fixed constant and define  $\bar{p} = \min\{p, 1-p\}$ , while  $c = c(n)$  and  $m = m(n) \geq n$  could be changing with  $n$ .

First, we define an induced subgraph that consists of  $k$  message vertices,  $kc$  client vertices as a  $k$ -pattern, if the following property is satisfied:

- each of the  $k$  messages is connected with  $c$  clients and each of the  $kc$  client is connected with only one message.

The  $k$ -pattern is shown in Fig. 1.

For a given random bipartite graph  $B$ , let us denote by  $Y_k$  the number of  $k$ -patterns. For an induced subgraph  $B'$  of  $B$ , let us denote by  $Y_k^{B'}$  the number of  $k$ -patterns on the subgraph  $B'$ .

We have the following calculation for the average number of  $k$ -patterns.

$$\mathbb{E}[Y_k] = \binom{m}{k} \binom{n}{kc} \binom{kc}{c, c, \dots, c} p^{kc} (1-p)^{k(k-1)c}. \quad (11)$$

It is easy to see that  $\mathbb{E}[Y_k]$  is decreasing with  $k$ , given other parameters fixed. Hence, we define  $k_0$  to be the maximum integer such that  $\mathbb{E}[Y_{k_0}] \geq 1$ , i.e.,  $k_0 = \max\{k | \mathbb{E}[Y_k] \geq 1\}$ . We next show that  $k_0$  is in the order of  $\log(n) + \frac{\log(m)}{c} - \log(c)$ . More accurately, we show in the following lemma.

**Lemma 5.**  $k_0$  satisfies:  $x_1 \leq k_0 \leq x_1 + O(1)$  for  $x_1 = 1 + \frac{1}{\log(1/(1-p))} [\log(n) + \frac{\log(m)}{c} - \log(c) - \frac{\log(\log(n) + \frac{\log(m)}{c} - \log(c))}{c} - \frac{\log \log(\frac{1}{1-p})}{c} + \log(p)] + o(1)$ .

*Proof.* Using the binomial inequality

$$\left(\frac{x}{y}\right)^y \leq \binom{x}{y} \leq \left(\frac{ex}{y}\right)^y, \quad (12)$$

we can bound the value of  $\mathbb{E}[Y_k]$  by

$$\begin{aligned} \mathbb{E}[Y_k] &\leq \left(\frac{em}{k}\right)^k \left(\frac{en}{c}\right)^c \left(\frac{e(n-c)}{c}\right)^c \left(\frac{e(n-2c)}{c}\right)^c \dots \left(\frac{e(n-kc)}{c}\right)^c p^{kc} (1-p)^{k(k-1)c} \\ &< \left(\frac{em}{k}\right)^k \left(\frac{en}{c}\right)^{kc} p^{kc} (1-p)^{k(k-1)c}, \end{aligned} \quad (13)$$

and

$$\begin{aligned} \mathbb{E}[Y_k] &\geq \left(\frac{m}{k}\right)^k \left(\frac{n}{c}\right)^c \left(\frac{n-c}{c}\right)^c \left(\frac{n-2c}{c}\right)^c \dots \left(\frac{n-kc}{c}\right)^c p^{kc} (1-p)^{k(k-1)c} \\ &> \left(\frac{m}{k}\right)^k \left(\frac{n-kc}{c}\right)^{kc} p^{kc} (1-p)^{k(k-1)c}. \end{aligned} \quad (14)$$

By taking  $\log(\cdot)$  on both sides, we get the following relationships:

$$\begin{aligned}\log(\mathbb{E}[Y_k]) &< k[1 + \log(m) - \log(k)] + kc[1 + \log(n) - \log(c)] + kc \log(p) + k(k-1)c \log(1-p), \\ \log(\mathbb{E}[Y_k]) &> k[\log(m) - \log(k)] + kc[\log(n - kc) - \log(c)] + kc \log(p) + k(k-1)c \log(1-p).\end{aligned}\tag{15}$$

Let us define two continuous functions  $f_1(x) = x[\log(m) - \log(x)] + xc[\log(n - xc) - \log(c)] + xc \log(p) + x(x-1)c \log(1-p)$  and  $f_2(x) = x[1 + \log(m) - \log(x)] + xc[1 + \log(n) - \log(c)] + xc \log(p) + x(x-1)c \log(1-p)$ . Hence, we can rewrite the above inequalities as  $f_1(k) < \log(\mathbb{E}[Y_k]) < f_2(k)$ . Note that the two functions  $f_1(x)$  and  $f_2(x)$  are monotonously decreasing around  $\log(n) + \frac{\log(m)}{c} - \log(c)$ .

Solving the equations  $f_1(x) = 0$  and  $f_2(x) = 0$ , we get

$$\begin{aligned}f_1(x_1) = 0, \text{ for } x_1 &= 1 + \frac{1}{\log(1/(1-p))} [\log(n) + \frac{\log(m)}{c} - \log(c) - \frac{\log(\log(n) + \frac{\log(m)}{c} - \log(c))}{c} \\ &\quad - \frac{\log \log(\frac{1}{1-p})}{t} + \log(p)] + o(1), \\ f_2(x_2) = 0, \text{ for } x_2 &= 1 + \frac{1}{\log(1/(1-p))} [\log(n) + \frac{\log(m)}{c} - \log(c) - \frac{\log(\log(n) + \frac{\log(m)}{c} - \log(c))}{c} \\ &\quad + 1 + \frac{1}{c} - \frac{\log \log(\frac{1}{1-p})}{c} + \log(p)] + o(1).\end{aligned}\tag{16}$$

We can see that both  $x_1$  and  $x_2$  are in the order of  $\log(n) + \frac{\log(m)}{c} - \log(c)$  and  $x_2 - x_1 = \frac{1}{\log(1/(1-p))} [1 + \frac{1}{c}] + o(1) \leq \frac{2}{\log(1/(1-p))} + o(1)$ , which is bounded by  $O(1)$ .

We also have  $\log(\mathbb{E}[Y_{\lfloor x_2 \rfloor}]) < f_2(\lfloor x_2 \rfloor) \leq f_2(x_2) = 0$  and  $\log(\mathbb{E}[Y_{\lfloor x_1 \rfloor}]) > f_1(\lfloor x_1 \rfloor) \geq f_1(x_1) = 0$ . This implies that  $x_1 - 1 < \lfloor x_1 \rfloor \leq k_0 \leq \lfloor x_2 \rfloor - 1 < x_2$ , from which the result follows.  $\square$

What we would like to show next is that the average number of  $k$ -patterns  $\mathbb{E}[Y_k]$  has the property that it changes fast around the value  $k_0$ . We have the following lemma.

**Lemma 6.**  $\mathbb{E}[Y_{k_1}]$  satisfies:  $\mathbb{E}[Y_{k_1}] \geq (\frac{n}{ec})^{3c(1+o(1))} m^{3(1+o(1))}$ , for  $k_1 = k_0 - 3$ .

*Proof.* We first have the following equation

$$\begin{aligned}\frac{\mathbb{E}[Y_{k_0-3}]}{\mathbb{E}[Y_{k_0}]} &= \frac{\binom{m}{k_0-3} \binom{n}{(k_0-3)c} \binom{(k_0-3)c}{c} \binom{(k_0-4)c}{c} \dots \binom{c}{c} p^{(k_0-3)c} (1-p)^{(k_0-3)(k_0-4)c}}{\binom{m}{k_0} \binom{n}{k_0 c} \binom{k_0 c}{c} \binom{(k_0-1)c}{c} \dots \binom{c}{c} p^{k_0 c} (1-p)^{k_0(k_0-1)c}} \\ &= \frac{k_0(k_0-1)(k_0-2)(c!)^3}{(m-k_0+3)(m-k_0+2)(m-k_0+1)(n-k_0c+1)(n-k_0c+2) \dots (n-k_0c+3c) p^{3c} (1-p)^{6k_0c-12c}} \\ &\geq \frac{(c!)^3}{m^3 n^{3c} (1-p)^{6c(k_0-2)}} \\ &\geq (\frac{n}{ec})^{3c(1+o(1))} m^{3(1+o(1))},\end{aligned}\tag{17}$$

where the last inequality follows from  $c! \geq e(c/e)^c$  and  $(1-p)^{6c(k_0-2)} = (\frac{nm^{1/c}}{c})^{6c(1+o(1))}$ , since  $k_0 - 2 = \frac{1}{\log(1/(1-p))} [\log(n) + \frac{\log(m)}{c} - \log(c) + o(\log(n) + \frac{\log(m)}{c} - \log(c))]$ .

Also note that  $\mathbb{E}[Y_{k_0}] \geq 1$  and the result follows from (17).  $\square$

Similarly, we can define  $k_0^{B'}$  as the maximum integer such that  $\mathbb{E}[Y_{k_0^{B'}}^{B'}] \geq 1$  and define  $k_1^{B'} = k_0^{B'} - 3$ .

Let us denote by  $\mathcal{B}(B, m', n')$  the family of all induced subgraphs of  $B$  by  $m'$  message vertices and  $n'$  client vertices. Next, we will discuss in different scenarios (in the following scenarios 1, 2, and 3) that we can find another integer  $\mathbb{K} = \mathbb{K}(m, n) = k_2^{B'} \leq k_1^{B'}$ , such that every induced subgraph  $B' \in \mathcal{B}(B, m', n')$  almost surely contains a  $\mathbb{K}$ -pattern. For the fourth scenario, we will discuss separately. The 4 scenarios with parameters  $\mathbb{K}$ ,  $m'$ , and  $n'$  are formally defined as follows.

**Definition 2.** We define the following 4 scenarios and how the corresponding parameters are related.

- *Scenario 1:*  $m < \exp(n^{1/15})$ . In this scenario, we set  $c = 1$ ,  $\mathbb{K} = \lfloor \frac{1}{\log(1/\bar{p})} [\log(m) - 3 \log \log(m) + 2 \log \log(\frac{1}{\bar{p}})] \rfloor = \Theta(\log(m))$ ,  $m' = \frac{m}{\log(m)}$ , and  $n' = \frac{n}{\log(m)}$ . If  $c > 1$ , we simply set  $c = 1$  and this is a stronger constraint.

- *Scenario 2:*  $m \geq \exp(n^{1/15})$ . In this scenario, we set  $c = 1$ ,  $\mathbb{K} = \lfloor \frac{1}{\log(1/\bar{p})} [\log(m) - 3 \log \log(m) + 2 \log \log(\frac{1}{\bar{p}})] \rfloor = \Theta(\log(m))$ ,  $m' = m - \mathbb{K} = m(1 - o(1))$ , and  $n' = \mathbb{K}$ . If  $c > 1$ , we simply set  $c = 1$  and this is a stronger constraint.

- *Scenario 3:*  $c = o(\frac{n^{1/7}}{\log^2(n)})$ . In this scenario, we set  $\mathbb{K} = \lfloor \frac{1}{\log(1/\bar{p})} [\log(n) - 3 \log \log(n) - 3 \log(c) + 2 \log \log(\frac{1}{\bar{p}})] \rfloor = \Theta(\log(n))$ ,  $m' = \frac{m}{\log(n)}$ , and  $n' = \frac{n}{c \log(n)}$ .

- *Scenario 4:*  $c = \Omega(\frac{n^{1/7}}{\log^2(n)})$ . In this scenario, we set  $\mathbb{K} = 1$ ,  $m' = 1$ , and  $n' = \frac{2c}{p}$ .

Note that the scenarios 1 and 2 are defined based on the relationship between  $m$  and  $n$ ; the scenarios 3 and 4 are defined based on the relationship between  $c$  and  $n$ . There may be overlaps between scenarios 1, 2 and scenarios 3, 4. We want to show the following lemma for the first 3 scenarios.

**Lemma 7.** For scenarios 1, 2, and 3 with parameters defined in Definition 2, every induced subgraph  $B' \in \mathcal{B}(B, m', n')$  almost surely contains a  $\mathbb{K}$ -pattern:

$$\Pr\{\exists B' \in \mathcal{B}(B, m', n'), \text{ s.t., } B' \text{ contains no } \mathbb{K}\text{-pattern}\} = o(1). \quad (18)$$

*Proof.* To prove this, we first use an “edge exposure” process to form a martingale based on the random subgraph  $B'$  [13], [16]. Specifically, we define  $X$  as a maximum number of  $\mathbb{K}$ -patterns in  $B'$  such that no two of them share a same message-client pair (i.e., any two  $\mathbb{K}$ -patterns either have no common message vertices or client vertices or both). We label the possible edges as  $1, 2, \dots, m'n'$  and denote by  $Z_l$  the random variable to indicate whether the edge  $l$  is exposed in the random graph, i.e.,  $Z_l = 1$  if the  $l$ -th possible edge is present in the graph and  $Z_l = 0$  otherwise. Therefore,  $X = f(Z_1, Z_2, \dots, Z_{m'n'})$  is a function of the variables  $Z_l$ . Define  $X_l = \mathbb{E}[X|Z_1, Z_2, \dots, Z_l]$  as a sequence of random variables for  $l = 1, 2, \dots, m'n'$ , then  $\{X_l\}$  is a Doob martingale and  $X_{m'n'} = X$ . Obviously, the function  $X = f(Z_1, Z_2, \dots, Z_{m'n'})$  is 1-Lipschitz, namely, flipping only one indicator function, some  $Z_l$ , the value of  $X$  differs by at most 1:  $|f(Z_1, \dots, Z_l, \dots, Z_{m'n'}) - f(Z_1, \dots, Z_{l-1}, 1 - Z_l, Z_{l+1}, \dots, Z_{m'n'})| \leq 1$ .

Note that the subgraph  $B'$  contains no  $\mathbb{K}$ -pattern is equivalent to  $X = 0$ . We then use the Azuma's inequality

$$\Pr\{\mathbb{E}[X] - X \geq a\} \leq \exp(-\frac{a^2}{2m'n'}), \text{ for } a > 0. \quad (19)$$

to get

$$\Pr\{X = 0\} = \Pr\{\mathbb{E}[X] - X \geq \mathbb{E}[X]\} \leq \exp(-\frac{\mathbb{E}^2[X]}{2m'n'}). \quad (20)$$

Hence, to bound  $\Pr\{X = 0\}$ , we only need to find a lower bound of  $\mathbb{E}[X]$ . We use the following probabilistic argument. For subgraph  $B'$ , we define  $\mathcal{K}$  as the family of all  $\mathbb{K}$ -patterns and  $\mathcal{P}$  as the family of all  $\mathbb{K}$ -pattern pairs that share at least a same message and a same client. Let us denote by  $B_1, B_2 \in \mathcal{B}(B', \mathbb{K}, \mathbb{K}_c)$  induced subgraphs of  $B'$  by  $\mathbb{K}$  message vertices and  $\mathbb{K}_c$  client vertices. Let us also denote by  $X_{B_1}$  and  $X_{B_2}$  the variables to indicate whether the subgraphs  $B_1$  and  $B_2$  are  $\mathbb{K}$ -patterns. Let us use the notation  $B_1 \sim B_2$  if two different subgraphs  $B_1$  and  $B_2$  share at least a same message vertex and a same client vertex. We then lower bound  $\mathbb{E}[X]$  using the following scheme for scenarios 1, 2, 3 (we will talk about how we bound  $\mathbb{E}[X]$  for scenario 4 later): randomly select a subset of  $\mathbb{K}$ -patterns from the set  $\mathcal{K}$  by picking up each  $\mathbb{K}$ -pattern with probability  $p^\dagger$  (the value of which we will determine later); if two selected  $\mathbb{K}$ -patterns  $B_1^\dagger$  and  $B_2^\dagger$  form a pair in the set  $\mathcal{P}$ , then remove one of them. Then,

$$\mathbb{E}[X] \geq p^\dagger \mathbb{E}[|\mathcal{K}|] - p^{\dagger 2} \mathbb{E}[|\mathcal{P}|], \quad (21)$$

where the first term in the expression is the average number of selected  $\mathbb{K}$ -patterns in  $\mathcal{K}$  and

the second term is the average number of  $\mathbb{K}$ -patterns that are removed because a pair in  $\mathcal{P}$  is selected with probability  $p^{\dagger 2}$ .

We observe that  $\mathbb{E}[|\mathcal{K}|] = \mathbb{E}[Y_{\mathbb{K}}^{B'}]$  and next we calculate  $\mathbb{E}[|\mathcal{P}|]$  and determine  $p^{\dagger}$ .

$$\begin{aligned}
\mathbb{E}[|\mathcal{P}|] &= \frac{1}{2} \sum_{B_1 \in \mathcal{B}(B', \mathbb{K}, \mathbb{K})} \sum_{B_2 \in \mathcal{B}(B', \mathbb{K}, \mathbb{K}) : B_2 \sim B_1} \mathbb{E}[X_{B_1} X_{B_2}] \\
&= \frac{1}{2} \sum_{B_1} \sum_{B_2 : B_2 \sim B_1} \Pr\{X_{B_1} = 1\} \Pr\{X_{B_2} = 1 | X_{B_1} = 1\} \\
&= \frac{1}{2} \binom{m'}{\mathbb{K}} \binom{n'}{\mathbb{K}} \mathbb{K}! \Pr\{X_{B_0} = 1\} \sum_{B_2 : B_2 \sim B_0} \Pr\{X_{B_2} = 1 | X_{B_0} = 1\} \\
&= \frac{1}{2} \mathbb{E}[Y_{\mathbb{K}}^{B'}] \sum_{B_2 : B_2 \sim B_0} \Pr\{X_{B_2} = 1 | X_{B_0} = 1\},
\end{aligned} \tag{22}$$

where the second equality is from the conditional probability formula, the third equality is by symmetry of the selection of  $B_1$  and we then take a fixed selection  $B_0$  consisting of the first  $\mathbb{K}$  messages and first  $\mathbb{K}$  clients.

Hence, we only need to calculate the term  $\sum_{B_2 : B_2 \sim B_0} \Pr\{X_{B_2} = 1 | X_{B_0} = 1\}$  for different scenarios. We upper bound this term from above by enumerating all subgraph  $B_2$  that has at least one common client vertex one common message vertex with  $B_0$ .

1) For scenario 1, we have

$$\sum_{B_2 : B_2 \sim B_0} \Pr\{X_{B_2} = 1 | X_{B_0} = 1\} \leq \sum_{j=1}^{\mathbb{K}} \sum_{i=1}^{\mathbb{K}} \binom{\mathbb{K}}{j} \binom{m'-\mathbb{K}}{\mathbb{K}-j} \binom{\mathbb{K}}{i} \binom{n'-\mathbb{K}}{\mathbb{K}-i} \mathbb{K}! \frac{p^{\mathbb{K}}(1-p)^{\mathbb{K}(\mathbb{K}-1)}}{\bar{p}^{ij}}, \tag{23}$$

where the inequality is because  $\bar{p}^{ij} \leq p^a(1-p)^{ij-a}$  for any non-negative integer  $a \leq ij$ . Let us define the term inside the summation as  $\Delta_{ij} \triangleq \binom{\mathbb{K}}{j} \binom{m'-\mathbb{K}}{\mathbb{K}-j} \binom{\mathbb{K}}{i} \binom{n'-\mathbb{K}}{\mathbb{K}-i} \mathbb{K}! \frac{p^{\mathbb{K}}(1-p)^{\mathbb{K}(\mathbb{K}-1)}}{\bar{p}^{ij}}$ .

We can see that for  $i = 1, 2, \dots, \mathbb{K}$  and  $j = 1, 2, \dots, \mathbb{K} - 1$ , we have

$$\begin{aligned}
\frac{\Delta_{i,j+1}}{\Delta_{i,j}} &= \frac{(\mathbb{K}-j)^2}{(j+1)(m'-2\mathbb{K}+j+1)} \bar{p}^{-i} \\
&\leq \frac{\mathbb{K}^2}{2(m'-2\mathbb{K}+2)} \bar{p}^{-\mathbb{K}} \\
&\leq \frac{\mathbb{K}^2}{m'} \bar{p}^{-\mathbb{K}} \\
&\leq \frac{\frac{1}{\log^2(1/\bar{p})} \log^2(m)}{m/\log(m)} \frac{m \log^2(1/\bar{p})}{\log^3(m)} \\
&\leq 1.
\end{aligned} \tag{24}$$

This implies that for all  $i = 1, 2, \dots, \mathbb{K}$  and  $j = 1, 2, \dots, \mathbb{K}$ ,  $\Delta_{i,j} \leq \Delta_{i,1}$ . Also note that for

$i = 1, 2, \dots, \mathbb{K} - 1,$

$$\begin{aligned}
\frac{\Delta_{i+1,1}}{\Delta_{i,1}} &= \frac{(\mathbb{K}-i)^2}{(i+1)(n'-2\mathbb{K}+i+1)} \bar{p}^{-1} \\
&\leq \frac{\mathbb{K}^2}{n' \bar{p}} \\
&\leq \frac{\log(m)^3}{n \bar{p} \log^2(1/\bar{p})} \\
&= o(1),
\end{aligned} \tag{25}$$

where the last equality holds for  $m < \exp(n^{1/10})$ . Hence,  $\Delta_{i,j} \leq \Delta_{1,1}$  for all  $i = 1, 2, \dots, \mathbb{K}$  and  $j = 1, 2, \dots, \mathbb{K}$ .

For  $\Delta_{1,1}$ , we have the following

$$\begin{aligned}
\frac{\Delta_{1,1}}{\mathbb{E}[Y_{\mathbb{K}}^{B'}]} &= \frac{\mathbb{K} \binom{m'-\mathbb{K}}{\mathbb{K}-1} \mathbb{K} \binom{n'-\mathbb{K}}{\mathbb{K}-1} \mathbb{K}! \frac{p^{\mathbb{K}} (1-p)^{\mathbb{K}(\mathbb{K}-1)}}{\bar{p}}}{\binom{m'}{\mathbb{K}} \binom{n'}{\mathbb{K}} \mathbb{K}! p^{\mathbb{K}} (1-p)^{\mathbb{K}(\mathbb{K}-1)}} \\
&= \frac{\mathbb{K}^4 (m'-\mathbb{K})! (m'-\mathbb{K})! (n'-\mathbb{K})! (n'-\mathbb{K})!}{\bar{p} m'! (m'-2\mathbb{K}+1)! n'! (n'-2\mathbb{K}+1)!} \\
&\leq \frac{\mathbb{K}^4 \log^2(m)}{\bar{p} m n} \leq \frac{\log^6(m)}{\bar{p} m n \log^4(1/\bar{p})}.
\end{aligned} \tag{26}$$

Plugging into (22), we have

$$\mathbb{E}[|\mathcal{P}|] \leq \frac{1}{2} \mathbb{E}^2[Y_{\mathbb{K}}^{B'}] \mathbb{K}^2 \frac{\log^6(m)}{\bar{p} \log^4(1/\bar{p}) m n} \leq \mathbb{E}^2[Y_{\mathbb{K}}^{B'}] \frac{\log^8(m)}{2 \bar{p} \log^6(1/\bar{p}) m n} \tag{27}$$

From Lemmas 5 and 6, we have  $k_0^{B'} = \frac{1}{\log(1/(1-p))} [\log(n) + \log(m) - 2 \log \log(m) - \log(\log(n) + \log(m) - 2 \log \log(m)) - \log \log(\frac{1}{1-p}) + \log(p)] + O(1)$  and  $\mathbb{E}[Y_{k_0^{B'}-3}^{B'}] \geq (\frac{mn}{e \log^2(m)})^{3(1+o(1))}$ . Obviously, we have  $\mathbb{K} < k_1^{B'} = k_0^{B'} - 3$  and  $\mathbb{E}[Y_{\mathbb{K}}^{B'}] \geq (\frac{mn}{e \log^2(m)})^{3(1+o(1))}$ . By setting the probability  $p^\dagger = \frac{\bar{p} \log^6(\frac{1}{\bar{p}}) m n}{\mathbb{E}[Y_{\mathbb{K}}^{B'}] \log^8(m)} < 1$ , we can bound the average number of  $X$ ,  $\mathbb{E}[X]$ , in e.q. (21), as

$$\mathbb{E}[X] \geq \frac{\bar{p}^2 \log^6(\frac{1}{\bar{p}}) m n}{2 \log^8(m)}. \tag{28}$$

Plugging (28) into (20), we can bound the following probability

$$\Pr\{B' \text{ contains no } \mathbb{K}\text{-pattern}\} \leq \exp\left(-\frac{\bar{p}^2 \log^{12}(1/\bar{p}) m n}{8 \log^{14}(m)}\right) \tag{29}$$

Therefore, we can bound the probability that any subgraph  $B'$  induced by  $m'$  messages and  $n'$  clients does not contain a  $\mathbb{K}$ -pattern:

$$\begin{aligned}
\Pr\{\exists B' \in \mathcal{B}(B, m', n'), s.t., B' \text{ contains no } \mathbb{K}\text{-pattern}\} &\leq \binom{m}{m'} \binom{n}{n'} \exp\left(-\frac{\bar{p}^2 \log^{12}(1/\bar{p}) m n}{8 \log^{14}(m)}\right) \\
&\leq 2^{m+n} \exp\left(-\frac{\bar{p}^2 \log^{12}(1/\bar{p}) m n}{8 \log^{14}(m)}\right) = o(1).
\end{aligned} \tag{30}$$

2) For scenario 2, we have

$$\sum_{B_2: B_2 \sim B_0} \Pr\{X_{B_2} = 1 | X_{B_0} = 1\} \leq \sum_{j=1}^{\mathbb{K}} \binom{\mathbb{K}}{\mathbb{K}-j} \binom{m'-\mathbb{K}}{\mathbb{K}-j} \mathbb{K}! \frac{p^{\mathbb{K}}(1-p)^{\mathbb{K}(\mathbb{K}-1)}}{\bar{p}^j \mathbb{K}}. \quad (31)$$

Let us define the term inside the summation as  $\Delta_j \triangleq \binom{\mathbb{K}}{\mathbb{K}-j} \binom{m'-\mathbb{K}}{\mathbb{K}-j} \mathbb{K}! \frac{p^{\mathbb{K}}(1-p)^{\mathbb{K}(\mathbb{K}-1)}}{\bar{p}^j \mathbb{K}}$ .

Then we can see that for  $j = 1, 2, \dots, \mathbb{K} - 1$ , we have

$$\begin{aligned} \frac{\Delta_{j+1}}{\Delta_j} &= \frac{(\mathbb{K}-j)^2}{(j+1)(m'-2\mathbb{K}+j+1)} \bar{p}^{-\mathbb{K}} \\ &\leq \frac{\mathbb{K}^2}{m'} \bar{p}^{-\mathbb{K}} \\ &\leq \frac{1}{\log^2(1/\bar{p})} \log^2(m) \frac{m \log^2(1/\bar{p})}{\log^3(m)} \\ &= o(1). \end{aligned} \quad (32)$$

This implies that for all  $j = 1, 2, \dots, \mathbb{K}$ ,  $\Delta_j \leq \Delta_1$ .

For  $\Delta_1$ , we have the following

$$\begin{aligned} \frac{\Delta_1}{\mathbb{E}[Y_{\mathbb{K}}^{B'}]} &= \frac{\mathbb{K} \binom{m'-\mathbb{K}}{\mathbb{K}-1} \mathbb{K}! \frac{p^{\mathbb{K}}(1-p)^{\mathbb{K}(\mathbb{K}-1)}}{\bar{p}}}{\binom{m'}{\mathbb{K}} \binom{n'}{\mathbb{K}} \mathbb{K}! p^{\mathbb{K}}(1-p)^{\mathbb{K}(\mathbb{K}-1)}} \\ &= \frac{\mathbb{K}(m'-\mathbb{K})!(m'-\mathbb{K})!}{\bar{p} m'!(m'-2\mathbb{K}+1)!} \\ &\leq \frac{\mathbb{K}^2}{\bar{p} m(1-o(1))} \leq \frac{\log^2(m)(1+o(1))}{\bar{p} m \log^2(1/\bar{p})}. \end{aligned} \quad (33)$$

Next, we have

$$\mathbb{E}[|\mathcal{P}|] \leq \frac{1}{2} \mathbb{E}^2[Y_{\mathbb{K}}^{B'}] \mathbb{K} \frac{\log^2(m)(1+o(1))}{\bar{p} \log^2(1/\bar{p}) m} \leq \mathbb{E}^2[Y_{\mathbb{K}}^{B'}] \frac{\log^3(m)}{2\bar{p} \log^3(1/\bar{p}) m} \quad (34)$$

From Lemmas 5 and 6, we have  $k_0^{B'} = \frac{1}{\log(1/(1-p))} [\log(\mathbb{K}) + \log(m - \mathbb{K}) - \log(\log(\mathbb{K}) + \log(m - \mathbb{K})) - \log \log(\frac{1}{1-p}) + \log(p)] + O(1) = \frac{1}{\log(1/(1-p))} [\log(\mathbb{K}) + \log(m - \mathbb{K}) - \log(\log(\mathbb{K}) + \log(m - \mathbb{K})) - \log \log(\frac{1}{1-p}) + \log(p)] + O(1) = \frac{1}{\log(1/(1-p))} [\log(m) - 2 \log \log(\frac{1}{1-p}) + \log(p)] + O(1)$  and  $\mathbb{E}[Y_{k_0^{B'}-3}^{B'}] \geq (\frac{m\mathbb{K}}{e})^{3(1+o(1))}$ . Obviously, we have  $\mathbb{K} < k_1^{B'} = k_0^{B'} - 3$  and  $\mathbb{E}[Y_{\mathbb{K}}^{B'}] \geq (\frac{m\mathbb{K}}{e})^{3(1+o(1))}$ . By setting the probability  $p^\dagger = \frac{\bar{p} \log^3(\frac{1}{\bar{p}}) m(1-o(1))}{\mathbb{E}[Y_{\mathbb{K}}^{B'}] \log^3(m)} < 1$ , we can bound the average number of  $X$ ,  $\mathbb{E}[X]$ , by

$$\mathbb{E}[X] \geq \frac{\bar{p} \log^3(\frac{1}{\bar{p}}) m(1-o(1))}{2 \log^3(m)}. \quad (35)$$

We then can bound the following probability using Azuma's inequality

$$\begin{aligned} \Pr\{B' \text{ contains no } \mathbb{K}\text{-pattern}\} &\leq \exp\left(-\frac{\bar{p}^2 \log^6(1/\bar{p}) m^2(1-o(1))}{8 \log^6(m) m' n'}\right) \\ &\leq \exp\left(-\frac{\bar{p}^2 \log^7(1/\bar{p}) m(1-o(1))}{8 \log^7(m)}\right). \end{aligned} \quad (36)$$



Therefore, we can bound the probability that any subgraph  $B'$  induced by  $m'$  messages and  $n'$  clients does not contain a  $\mathbb{K}$ -pattern:

$$\begin{aligned} \Pr\{\exists B' \in \mathcal{B}(B, m', n'), s.t., B' \text{ contains no } \mathbb{K}\text{-pattern}\} &\leq \binom{m}{m'} \binom{n}{n'} \exp\left(-\frac{\bar{p}^2 \log^7(1/\bar{p}) m(1-o(1))}{8 \log^7(m)}\right) \\ &\leq m^n 2^n \exp\left(-\frac{\bar{p}^2 \log^7(1/\bar{p}) m(1-o(1))}{8 \log^7(m)}\right) = o(1), \end{aligned} \quad (37)$$

where the last equality follows from that  $n \leq \log^{15}(m)$ .

3) For scenario 3, we have

$$\sum_{B_2: B_2 \sim B_0} \Pr\{X_{B_2} = 1 | X_{B_0} = 1\} \leq \sum_{j=1}^{\mathbb{K}} \sum_{i=1}^{\mathbb{K}c} \binom{\mathbb{K}}{j} \binom{m'-\mathbb{K}}{\mathbb{K}-j} \binom{\mathbb{K}c}{i} \binom{n'-\mathbb{K}c}{\mathbb{K}c-i} \binom{\mathbb{K}c}{c,c,\dots,c} \frac{p^{\mathbb{K}c(1-p)^{\mathbb{K}c(\mathbb{K}-1)}}}{\bar{p}^{ij}}, \quad (38)$$

Let us define the term inside the summation as  $\Delta_{i,j} \triangleq \binom{\mathbb{K}}{j} \binom{m'-\mathbb{K}}{\mathbb{K}-j} \binom{\mathbb{K}c}{i} \binom{n'-\mathbb{K}c}{\mathbb{K}c-i} \binom{\mathbb{K}c}{c,c,\dots,c} \frac{p^{\mathbb{K}c(1-p)^{\mathbb{K}c(\mathbb{K}-1)}}}{\bar{p}^{ij}}$ .

Then we can see that for  $j = 1, 2, \dots, \mathbb{K}$  and  $i = 1, 2, \dots, \mathbb{K}c - 1$ , we have

$$\begin{aligned} \frac{\Delta_{i+1,j}}{\Delta_{i,j}} &= \frac{(\mathbb{K}c-i)^2}{(i+1)(n'-2\mathbb{K}c+i+1)} \bar{p}^{-j} \\ &\leq \frac{\mathbb{K}^2 c^2}{n'} \bar{p}^{-\mathbb{K}} \\ &\leq \frac{\frac{1}{\log^2(1/\bar{p})} c^3 \log(n)}{n} \frac{n \log^2(1/\bar{p})}{c^3 \log^3(n)} \\ &\leq 1. \end{aligned} \quad (39)$$

This implies that for all  $i = 1, 2, \dots, \mathbb{K}c$ ,  $\Delta_{i,j} \leq \Delta_{1,j}$ .

We also note that for  $j = 1, 2, \dots, \mathbb{K} - 1$ , we have

$$\begin{aligned} \frac{\Delta_{1,j+1}}{\Delta_{1,j}} &= \frac{(\mathbb{K}-j)^2}{(j+1)(m'-2\mathbb{K}+j+1)} \bar{p}^{-1} \\ &\leq \frac{\mathbb{K}^2}{2\bar{p}(m'-2\mathbb{K}+2)} \\ &\leq \frac{\mathbb{K}^2}{\bar{p}m'} \\ &\leq \frac{\log^3(n)}{\bar{p} \log^2(1/\bar{p})m} \\ &= o(1). \end{aligned} \quad (40)$$

This implies that for all  $i = 1, 2, \dots, \mathbb{K}c$  and  $j = 1, 2, \dots, \mathbb{K}$ ,  $\Delta_{i,j} \leq \Delta_{1,j} \leq \Delta_{1,1}$ .

For  $\Delta_{1,1}$ , we have the following

$$\begin{aligned} \frac{\Delta_{1,1}}{\mathbb{E}[Y_{\mathbb{K}}^{B'}]} &= \frac{\mathbb{K} \binom{m'-\mathbb{K}}{\mathbb{K}-1} \mathbb{K} c \binom{n'-\mathbb{K}c}{\mathbb{K}c-1} \binom{\mathbb{K}c}{c,c,\dots,c} \frac{p^{\mathbb{K}c(1-p)} \mathbb{K} c (\mathbb{K}-1)}{p} \\ &\leq \frac{\mathbb{K}^4 c^2}{\bar{p} m' n'} \\ &\leq \frac{c^3 \log^6(n)}{\bar{p} m n \log^4(1/\bar{p})}. \end{aligned} \quad (41)$$

Next, we have

$$\mathbb{E}[|\mathcal{P}|] \leq \frac{1}{2} \mathbb{E}^2[Y_{\mathbb{K}}^{B'}] \mathbb{K}^2 c \frac{c^3 \log^6(n)}{\bar{p} m n \log^4(1/\bar{p})} \leq \mathbb{E}^2[Y_{\mathbb{K}}^{B'}] \frac{c^4 \log^8(n)}{2\bar{p} \log^6(1/\bar{p}) m n} \quad (42)$$

From Lemmas 5 and 6, we have  $k_0^{B'} = \frac{1}{\log(1/(1-p))} [\log(n) + \frac{\log(m)}{c} - 2 \log(c) - (1+1/c) \log \log(n) - \frac{\log[\log(n) + \log(m)/c - 2 \log(c)]}{c} - \frac{\log \log(1/(1-p))}{c} + \log(p)] + O(1)$  and  $\mathbb{E}[Y_{k_0^{B'}-3}^{B'}] \geq (\frac{n}{ec^2 \log(n)})^{3c(1+o(1))} (\frac{m}{\log(n)})^{3(1+o(1))}$ . Obviously, we have  $\mathbb{K} < k_1^{B'} = k_0^{B'} - 3$  and  $\mathbb{E}[Y_{\mathbb{K}}^{B'}] \geq (\frac{n}{ec^2 \log(n)})^{3c(1+o(1))} (\frac{m}{\log(n)})^{3(1+o(1))}$ . By setting the probability  $p^\dagger = \frac{\bar{p} \log^6(\frac{1}{\bar{p}}) m n}{\mathbb{E}[Y_{\mathbb{K}}^{B'}] c^4 \log^8(n)} < 1$ , we can bound the average number of  $X$ ,  $\mathbb{E}[X]$ , by

$$\mathbb{E}[X] \geq \frac{\bar{p} \log^6(\frac{1}{\bar{p}}) m n}{2c^4 \log^8(n)}. \quad (43)$$

We then can bound the following probability using Azuma's inequality

$$\begin{aligned} \Pr\{B' \text{ contains no } \mathbb{K}\text{-pattern}\} &\leq \exp\left(-\frac{\bar{p}^2 \log^{12}(1/\bar{p}) m^2 n^2}{8c^8 \log^{16}(n) m' n'}\right) \\ &\leq \exp\left(-\frac{\bar{p}^2 \log^{12}(1/\bar{p}) m n}{8c^7 \log^{14}(n)}\right). \end{aligned} \quad (44)$$

Therefore, we can bound the probability that any subgraph  $B'$  induced by  $m'$  messages and  $n'$  clients does not contain a  $\mathbb{K}$ -pattern:

$$\begin{aligned} \Pr\{\exists B' \in \mathcal{B}(B, m', n'), s.t., B' \text{ contains no } \mathbb{K}\text{-pattern}\} &\leq \binom{m}{m'} \binom{n}{n'} \exp\left(-\frac{\bar{p}^2 \log^{12}(1/\bar{p}) m n}{8c^7 \log^{14}(n)}\right) \\ &\leq 2^{m+n} \exp\left(-\frac{\bar{p}^2 \log^{12}(1/\bar{p}) m n}{8c^7 \log^{14}(n)}\right) = o(1), \end{aligned} \quad (45)$$

where the last equality follows from that  $c = o(\frac{n^{1/7}}{\log^2(n)})$ .  $\square$

We then can reiterate Theorem 2 in a slightly different way.

**Theorem 2'.** *The number of broadcast transmissions for random graph instance  $B(m, n, c, p)$  is almost surely upper bounded by*

- $O(\frac{n}{\log(m)})$ , for any  $c \geq 1$ ;
- $O(\frac{n}{c \log(n)})$ , for  $c = o(\frac{n^{1/7}}{\log^2(n)})$ ;

- $O(\frac{n}{c} + \log(c))$ , for  $c = \Omega(\frac{n^{1/7}}{\log^2(n)})$ .

*Proof.* For scenarios 1, 2, and 3, we can proceed using the following transmission scheme.

- Start from the original bipartite graph representation  $B$ . If there are more than  $m'$  messages and  $n'$  clients in the graph, we pick a  $\mathbb{K}$ -pattern to encode the messages and make one transmission. We remove the satisfied clients and the used messages. If there are less than  $n'$  clients, we can almost surely use  $n'$  transmissions to satisfy the remaining clients, which follows from that a subgraph of  $B$  that contains  $n'$  vertices on both sides almost surely have a perfect matching [17].

From Lemma 7, we can see that the above scheme can be done almost surely. Hence, we can almost surely use the number of transmissions  $\frac{n}{\mathbb{K}c} + n'$ , from which the first two parts follow.

Now, let us prove the third part of the theorem for scenario 4 with  $c = \Omega(\frac{n^{1/7}}{\log^2(n)})$ . We use a slightly different but simple proof technique. By setting  $n' = \frac{2c}{p}$ , we use a 2-step transmission scheme.

- In the first step, we arbitrarily make  $n/c$  uncoded transmissions. After each transmission, we remove up to  $c$  satisfied clients as many as possible.
- In the second step, we divide the remaining unsatisfied clients into groups as few as possible, each with up to  $c$  clients, we use pliable index coding scheme to satisfy each of the groups.

We want to show that we can almost surely satisfy at least  $n - n'$  clients by using these  $n/c$  uncoded transmissions in the first step. Hence, we can almost surely divide the remaining unsatisfied clients into at most  $n'/c = 2/p$  groups and these groups almost surely take  $\frac{2}{p}O(\log(c))$  broadcast transmissions [10].

For a fixed uncoded transmission, e.g., message  $b_j$ , we would like to show that the probability that this transmission cannot satisfy  $c$  clients is exponentially small if the remaining unsatisfied clients is more than  $n'$ . Let us denote by  $D$  the number of connections for message vertex  $b_j$  to any  $n'$  remaining client vertices. Then obviously,  $\mathbb{E}[D] = n'p = 2c$  and the probability that the uncoded transmission of  $b_j$  cannot satisfy  $c$  clients (i.e., a 1-pattern exists) can be bounded by the following Chernoff bound:

$$\begin{aligned} \Pr\{b_j \text{ cannot satisfy } c \text{ clients}\} &\leq \Pr\{D \leq c\} = \Pr\{D \leq (1 - 1/2)\mathbb{E}[D]\} \\ &\leq \exp(-\frac{2c(1/2)^2}{2}) = \exp(-\frac{c}{4}). \end{aligned} \tag{46}$$

After  $n/c$  uncoded transmissions, the probability that the number of remaining unsatisfied

clients is more than  $n'$  can be bounded as follows:

$$\begin{aligned} & \Pr\{n/c \text{ uncoded transmissions cannot satisfy } n - n' \text{ clients}\} \\ & \leq \frac{n}{c} \exp(-\frac{c}{4}) \leq n^{6/7} \log^2(n) o(1) \exp(-\frac{n^{1/7} \Omega(1)}{4 \log^2(n)}) = o(1). \end{aligned} \quad (47)$$

Combining the two steps, we have the number of broadcast transmissions almost surely upper bounded by  $n/c + \frac{2}{p} O(\log(c)) = O(n/c + \log(c))$  for scenario 4.  $\square$

Note that for Theorem 2', we can combine the results and have the number of broadcast transmissions almost surely upper bounded by  $O(\min\{\frac{n}{\log(m)}, \frac{n}{c \log(n)}\})$  for  $c = o(\frac{n^{1/7}}{\log^2(n)})$  and  $O(\min\{\frac{n}{\log(m)}, \frac{n}{c} + \log(c)\})$  for  $c = \Omega(\frac{n^{1/7}}{\log^2(n)})$ .

## APPENDIX C

### PROPERTIES OF PLIABLE INDEX CODING BASED SHUFFLING

#### A. Hamming Distance Analysis

We analyze the Hamming distance of our pliable index coding based shuffling. We first note that across different worker nodes, the Hamming distance is at least  $2(s - m_1 + m_1/r)$ , as in the outer layer of the transmission structure, two different worker nodes have common messages in no more than one group.

Next, we evaluate the Hamming distance across iterations for the same worker  $i$ . Let us define a *truncated cache state* on group  $g$  for worker  $i$  at iteration  $t$ ,  $z_i^t|_g \in \{0, 1\}^{m_1}$ , as a  $m_1$ -tuple that consists of coordinates of  $z_i^t$  corresponding to messages in group  $g$ . We first consider the Hamming distance  $H|_g$  between truncated cache state on a specific group  $g$  for worker  $i$  across iterations. We claim that the average Hamming distance  $H|_g$  across all iterations is at least the average Hamming Distance between two consecutive iterations, i.e., for two given iterations  $t_1 < t_2$ ,  $\Pr\{z_i^{t_1}|_g = z_i^{t_2}|_g\} \leq \Pr\{z_i^{t_1}|_g = z_i^{t_1+1}|_g\}$ .

To prove this, we use a random walk model on a graph  $G(V, E)$  that is constructed as follows. Each vertex  $v \in V$  corresponds to one of  $\binom{m_1}{m_1(1-1/r)}$  possible *truncated cache states*  $z_i^t|_g$ , or state for short, i.e., all binary vectors of length  $m_1$  and weight  $m_1(1 - 1/r)$ . There is an edge between two states  $v_1$  and  $v_2$  if and only if their Hamming distance is no more than 2, i.e., each vertex  $v$  has a self-loop and there is an edge connecting two vertices of Hamming distance 2. Thus, a vertex  $v$  has  $m_1^2(1/r - 1/r^2)$  connections with other vertices. Originally, worker  $i$  is in any of the  $\binom{m_1}{m_1(1-1/r)}$  possible states with equal probability. Using our proposed shuffling scheme, after each iteration, worker  $i$  remains in the same state with probability  $1 - p_1 \leq 1 - 1/e$  ( $p_1$

is defined as the probability that a worker can decode a new message during each transmission) and changes to a neighboring state with probability  $\frac{p_1}{em_1^2(1/r-1/r^2)}$  according to Lemma 3. Assume at iterations  $t_1$ , worker  $i$  is in some state  $v_1 \in V$ . At iteration  $t_2$ , worker  $i$ 's state is a random variable with some distribution. Let us denote by  $p_v^t$  the probability that worker  $i$  is in state  $v$  at iteration  $t$ . Then we have the flow conservation equation:

$$\begin{aligned}
p_{v_1}^{t_2} &= p_{v_1}^{t_2-1}(1 - p_1) + \sum_{v \neq v_1: \{v, v_1\} \in E} p_v^{t_2-1} \frac{p_1}{m_1^2(1/r-1/r^2)} \\
&= p_{v_1}^{t_2-1}(1 - p_1) + p_{v_0}^{t_2-1} \frac{p_1}{m_1^2(1/r-1/r^2)} m_1^2(1/r - 1/r^2) \\
&\leq p_{v_1}^{t_2-1}(1 - p_1) + \frac{1-p_{v_1}^{t_2-1}}{m_1^2(1/r-1/r^2)} p_1 \\
&\leq 1 - p_1 = p_{v_1}^{t_1+1},
\end{aligned} \tag{48}$$

where the second equality holds because the probabilities for worker  $i$  in  $v_1$ 's neighbors,  $p_v^{t_2-1}$  for  $v \neq v_1: \{v, v_1\} \in E$ , are all equal by symmetry, and thus we can pick a fixed neighbor  $v_0$  of  $v_1$ ; the first inequality holds because  $p_{v_0}^{t_2-1}$  is at most  $\frac{1-p_{v_1}^{t_2-1}}{m_1^2(1/r-1/r^2)}$ , i.e., worker  $i$  has equal probability in any of  $v_1$ 's neighbors by symmetry and the probability that worker  $i$  is in one of  $v_1$ 's neighbors is at most  $1 - p_{v_1}^{t_2-1}$ ; and the second inequality holds because the function  $g(p_{v_1}^{t_2-1}) = p_{v_1}^{t_2-1}(1 - p_1) + \frac{1-p_{v_1}^{t_2-1}}{m_1^2(1/r-1/r^2)} p_1$  is an increasing function and achieves the maximum for  $p_{v_1}^{t_2-1} = 1$ . Our claim is proved.

Hence, the average Hamming distance  $H|_g$  can be lower bounded by:

$$H|_g \geq 0 \cdot (1 - \frac{1}{e}) + 2 \cdot \frac{1}{e} = 2/e. \tag{49}$$

We then consider the average Hamming distance across all the groups in  $D(i)$ . Since  $|D(i)| = \frac{s}{m_1(1-1/r)}$ , this is at least  $\frac{s}{m_1(1-1/r)} 2/e = \frac{2s}{em_1(1-1/r)}$ . Therefore, on average  $H \geq \min\{\frac{2s}{em_1(1-1/r)}, 2(s - m_1 + m_1/r)\}$ .

### B. Independence and Randomness Preserving Property

Originally, if the worker nodes in  $N(g)$  have independently and uniformly at random cached  $m_1(1 - 1/r)$  messages in group  $g$ , then we observe that the pliable index coding based shuffling scheme maintains this ‘‘independence and randomness’’ property. Without loss of generality, assume the worker nodes in  $N(g)$  are  $1, 2, \dots, n_1$ , where  $n_1 = |N(g)|$ . Again, we use the graph constructed above.

**Corollary 1.** *The pliable index coding based shuffling scheme maintains the “independence and randomness” property. Formally, if the following two properties hold for iteration  $t$ :*

$$\Pr\{z_1^t|_g = v_1, z_2^t|_g = v_2, \dots, z_{n_1}^t|_g = v_{n_1}\} = \Pr\{z_1^t|_g = v_1\} \Pr\{z_2^t|_g = v_2\} \dots \Pr\{z_{n_1}^t|_g = v_{n_1}\}, \quad (50)$$

*for any state tuple  $(v_1, v_2, \dots, v_{n_1}) \in V^{n_1}$ , and*

$$\Pr\{z_i^t|_g = v_i\} = \frac{1}{|V|}, \quad (51)$$

*for any worker  $i \in [n_1]$  and state  $v_i \in V$ ; then these two properties also hold for iteration  $t+1$ :*

$$\begin{aligned} & \Pr\{z_1^{t+1}|_g = v'_1, z_2^{t+1}|_g = v'_2, \dots, z_{n_1}^{t+1}|_g = v'_{n_1}\} \\ &= \Pr\{z_1^{t+1}|_g = v'_1\} \Pr\{z_2^{t+1}|_g = v'_2\} \dots \Pr\{z_{n_1}^{t+1}|_g = v'_{n_1}\}, \end{aligned} \quad (52)$$

*for any state tuple  $(v'_1, v'_2, \dots, v'_{n_1}) \in V^{n_1}$ , and*

$$\Pr\{z_i^{t+1}|_g = v'_i\} = \frac{1}{|V|}, \quad (53)$$

*for any worker  $i \in [n_1]$  and state  $v_i \in V$ .*

*Proof.* The second property is obvious. Indeed, by symmetry of the constructed graph, if worker  $i$  is in every state with equal probability, then after one iteration (one random walk), worker  $i$  remains in every state with equal probability.

We then show the first property. We have the following

$$\begin{aligned} & \Pr\{z_1^{t+1}|_g = v'_1, \dots, z_{n_1}^{t+1}|_g = v'_{n_1}\} \\ &= \sum_{(v_1, \dots, v_{n_1})} \Pr\{z_1^t|_g = v_1, \dots, z_{n_1}^t|_g = v_{n_1}\} \cdot \\ & \quad \Pr\{z_1^{t+1}|_g = v'_1, \dots, z_{n_1}^{t+1}|_g = v'_{n_1} \mid z_1^t|_g = v_1, \dots, z_{n_1}^t|_g = v_{n_1}\} \quad (54) \\ &= \frac{n_1}{|V|} \sum_{(v_1, \dots, v_{n_1})} \Pr\{z_1^{t+1}|_g = v_1, \dots, z_{n_1}^{t+1}|_g = v_{n_1} \mid z_1^t|_g = v'_1, \dots, z_{n_1}^t|_g = v'_{n_1}\} \\ &= \frac{n_1}{|V|}, \end{aligned}$$

where the first equality holds due to the total probability theorem; the second equality holds because of the initial two properties for iteration  $t$ , i.e., e.q. (50) and (51), and the “reversibility

property” of the random walk, i.e.,

$$\begin{aligned} & \Pr\{z_1^{t+1}|_g = v'_1, \dots, z_{n_1}^{t+1}|_g = v'_{n_1} \mid z_1^t|_g = v_1, \dots, z_{n_1}^t|_g = v_{n_1}\} \\ &= \Pr\{z_1^{t+1}|_g = v_1, \dots, z_{n_1}^{t+1}|_g = v_{n_1} \mid z_1^t|_g = v'_1, \dots, z_{n_1}^t|_g = v'_{n_1}\}. \end{aligned} \quad (55)$$

The “reversibility property” describes that the probability walking from  $(v_1, \dots, v_{n_1})$  to  $(v'_1, \dots, v'_{n_1})$  is equal to that of walking from  $(v'_1, \dots, v'_{n_1})$  to  $(v_1, \dots, v_{n_1})$ . Indeed, if we use the same coded transmission and a reverse discarding process, then we achieve the goal. For example, if a worker has messages  $\{1, 2, 3\}$  in its cache, and the transmission is  $b_1 + b_2 + b_3 + b_4$ ; then the worker decodes message 4 and replaces message 1 and at last has cached messages  $\{2, 3, 4\}$ . If we reverse the process, we start from messages  $\{2, 3, 4\}$  in cache; using the same transmission  $b_1 + b_2 + b_3 + b_4$ , the worker decodes  $b_1$  and replaces message 4, resulting in cached messages  $\{1, 2, 3\}$ . This can be done with equal probability across all workers. The corollary is proved.  $\square$